Project Proposal: Improving Q-Learning by Probability States Search

Aoyun Zhang Zhanghan Ke

1. Introduction

Q-Learning is an algorithm to train an agent to play the game in a certain environment by a Q-table. In the learning process, an action is chosen in the current state to observe the outcome state and reward after taking the action, then the Q-table is updated by the reward. By this way, agent could learn how to gain a higher score in the game.

But one main problem in Q-Learning is that it converges very slowly. We think two parts of the algorithm may cause it. First, the default action selection strategy of Q-Learning is ε-greedy, which selects action randomly at a ratio of ε to balance between exploration and exploitation. However, the state with random exploration may be recorded before, therefore the ε-greedy will explores some repeated states, which is very ineffective for searching. Second, at the beginning stage, Q-table be updated by unreliable value since the agent is unable to choose the right action, it also causes the problem for convergence.

In this project, we will use probability estimation to determine whether a state need to be explored or only used history Q-value, and also try to improve the action selection strategy. We will also verify our improved algorithm in a famous game, Flappy Bird, to show the performance.

2. Our Plan

2.1 Algorithm design

Here we assume a replay dataset *D* including some "right" key-value samples, (state, action). The "right" means if the agent follows the corresponding action of the recorded state, it will always get positive reward.

For the first problem, we can estimate a probability distribution *Pi* for each action *Ai* using *D*. For each new state, if the computed probability of one action higher than the threshold (a hyperparameter), means in a high probability this state is known, we could choose the action by Q-values directly, otherwise we will choose an action randomly since the state never be explored before, such strategy could help agent explore new states more efficient.

For the second problem, when we choose the action from Q-table for state *s*, we can multiply the probability *Pi(s)* to *Q(s, Ai)* as a priori to help to select the next action. As we said above, in the beginning stage, Q-table is unreliable but *Pi* from a "right" *D*, such operation could balance the Q-value for each action to make a more accurate decision.

2.2 Data capturing

One problem in our method is how to get the "right" replay dataset *D* for probability estimation. To address this problem, we define a "*great round"* for the game which have a fixed time interval to judge the game, e.g. in Flappy Bird, interval between two pipes could be a round, and the "great round" means bird not died in this round, so the frames in "great round" must be "right". We can store data of these "great round" as *D* to train our model. However, if the game without fixed interval, the only way to get *D* is playing it for several times by human.

3. Evaluation Criterion

The goal of us is to improve the Q-Learning algorithm. We will evaluate our method on two aspects:

(1) Training original Q-Learning and our improved version in a relatively short time, i.e. playing fewer episodes than normal, and compare the convergence by the test mark.

(2) Training original Q-Learning and our improved version in enough time, i.e. playing enough episodes to make sure both algorithms are converged, and compare the final performance by the test mark.